7网络安全

2019年6月20日 15:18

-
- 网络安全问题概述

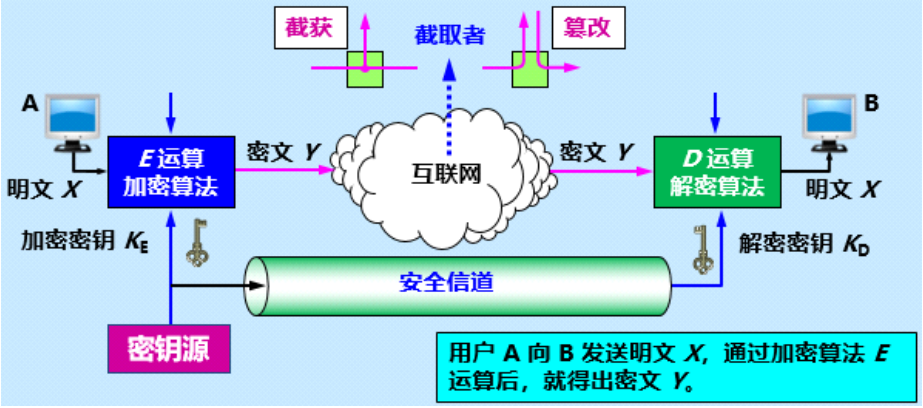
1- 计算机网络面临的安全性威胁

1. 被动攻击/窃听/截获：观察和分析协议数据单元PDU，但不干扰信息
   (1) 一般不易理解内部信息，但可以观察协议控制信息，了解通信实体地址和身份，研究PDU长度和通信频度，因而又被称traffic analysis流量分析
   (2) 战争时期，发现某处大量异常通信量就能猜到指挥所位置
2. 主动攻击
   (1) 篡改：修改报文流
   (2) rogue program恶意程序：
      1) virus病毒：修改其他程序，把自己或变种复制进去“传染”
      2) worm蠕虫：用通信功能将自己发到其他结点并自启动
      3) Trojan horse特洛伊木马：藏在其他程序里，再偷偷复制出来
      4) logic bomb逻辑炸弹：定时删去系统文件
      5) backdoor knocking后门入侵：用系统漏洞，通过网络入侵系统
      6) 流氓软件：强制安装，难卸载，劫持浏览器，弹广告，收集用户信息，恶意卸载，恶意捆绑等
   (3) Denial of Service拒绝服务DoS/网络带宽攻击/连通性攻击：向服务器发送大量分组，直至瘫痪。若互联网上不同计算机攻击同一网站，又称Distributed分布式拒绝服务
   (4) 交换机中毒poisoning：伪造不存在的MAC地址，使交换表满
3. 被动攻击难检测，但主动攻击可采取措施检测，因此，网络安全的目标：
   (1) 防止被分析出报文内容和流量
   (2) 防止恶意程序
   (3) 检测篡改和拒绝服务
4. 实现方法：加密解决被动攻击，加密和鉴别解决主动攻击

2- 安全的计算机网络

1. 保密性：只有发送接收方看得懂
2. 端点鉴别：能确定发送接收方的真实身份
3. 信息完整：不被篡改
4. 运行安全：access control访问控制权限，尤其在multilevel security多级安全下更重要

3- 数据加密模型

1. 

(1) 加密encryption；解密deciphering

(2) 如果不论截取者获得了多少密文，但在密文中都没有足够的信息来唯一地确定出对应的明文，则这一密码体制称为无条件安全的，或称为理论上是不可破的

(3) 如果密码体制中的密码不能被可使用的计算资源破译，则这一密码体制称为在计算上是安全的

- **加密和解密用的密钥*K*(key)是一串秘密的字符串（即比特串）。**
- **明文通过加密算法 *E* 和加密密钥 *K* 变成密文：**

$$Y = E_K(X) \qquad (7\text{-}1)$$

2.
- **接收端利用解密算法 *D* 运算和解密密钥 *K* 解出明文 *X*。解密算法是加密算法的逆运算。**

$$D_K(Y) = D_K(E_K(X)) = X \qquad (7\text{-}2)$$

- **加密密钥和解密密钥可以一样，也可以不一样。**
- **密钥通常由密钥中心提供。**
- **当密钥需要向远地传送时，一定要通过另一个安全信道。**

(1) cryptography密码编码学是密码体制的设计学。

(2) cryptanalysis密码分析学则是在未知密钥的情况下从密文推演出明文或密钥的技术

(3) cryptology密码编码学与密码分析学合起来即为密码学

-
- 两类密码体制

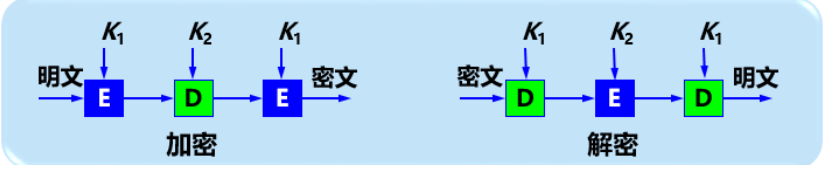
1- 常规/对称密钥密码体制：加密密钥与解密密钥是相同的密码体制

1. Data Encryption Standard数据加密标准DES：保密性仅取决于密钥的保密，算法是公开的

(1) 每64位一组，处理加密，实际密钥长度56位，8位用于奇偶校验

(2) triple DES：用一个密钥加密，再用另一个密钥解密，然后再使用第一个密钥加密，即

$$Y = DES_{K1}(DES^{-1}_{K2}(DES_{K1}(X))) \qquad (7\text{-}3)$$

1) 

(3) 广泛用于网络、金融、信用卡等

(4) NIST最终选择比利时的Rijndael算法为Advanced Encryption

StandardAES，以取代DES

2- 公钥密码体制：使用不同的加密密钥与解密密钥

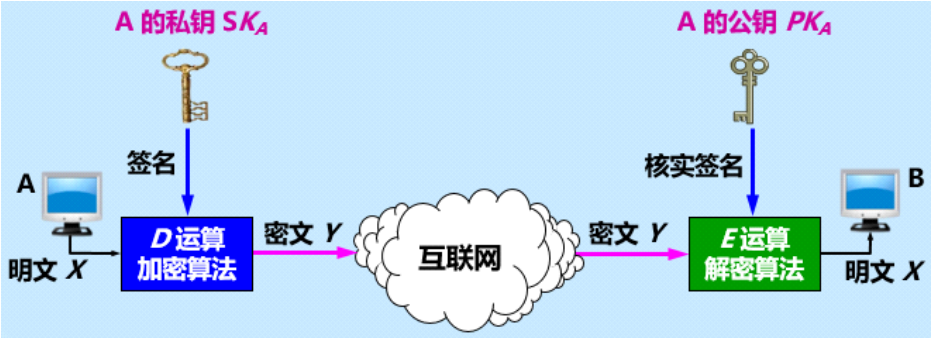
1. 是“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制
2. 加密算法E和解密算法D都是公开的，加密密钥也是public key公钥，而解密密钥是secret key私钥/秘钥，是唯一需要保密的（注意，根据公钥猜不到秘钥）
   (1) 用公钥加密，私钥解密，即实现了多对一保密
   (2) 用私钥加密，公钥解密，及实现了数字签名
   (3) 避免大量密钥分配的问题，和实现对数字签名的要求是公钥体制的出现原因
3. 因为加密方法安全性仅取决于密钥长度及攻破所需计算量
   (1) 公钥体制算法开销较大，密钥分配协议也很麻烦
4. 加密方法：发送者 A 用 B 的公钥 PKB 对明文 X 加密（E 运算）后，接收者 B 用自己的私钥 SKB 解密（D 运算），即可恢复出明文：
   (1) $D_{SK_B}(Y)=D_{SK_B}(E_{PK_B}(X))=X$
   (2) 注意加密密钥不能用来解密
   (3) $D_{PK_B}(E_{PK_B}(X)) \neq X$
   (4) 但加密和解密作为互逆运算，可以调换顺序
   (5) $E_{PK_B}(D_{SK_B}(X))=D_{SK_B}(E_{PK_B}(X))=X$

-
- 数字签名

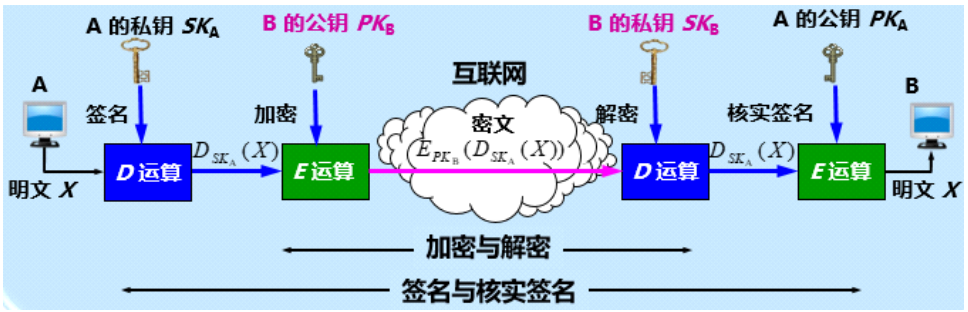
1- 数字签名目的

1. 报文鉴别——接收者能够核实发送者对报文的签名（证明来源）；
2. 报文的完整性——发送者事后不能抵赖对报文的签名（防否认）；
3. 不可否认——接收者不能伪造对报文的签名（防伪造）

2- 基于公钥的简单数字签名

1. 鉴别来源：除 A 外没有别人能具有 A 的私钥，所以没有别人能产生这个密文。因此 B 相信报文 X 是 A 签名发送的
2. 内容完整：若 A 要抵赖曾发送报文给 B，B 可将明文和对应的密文出示给第三者。第三者很容易用 A 的公钥去证实 A 确实发送 X 给 B。
3. 不可否认：若 B 将 X 伪造成 X'，则 B 不能在第三者前出示对应的密文。这样就证明了 B 伪造了报文
4. 

3- 具有保密性的基于公钥的数字签名

1. 用接收方的公钥做第二次加密
2. 

-
- 鉴别

1. Authentication鉴别：
   (1) 实体鉴别：验证通信对象非冒充者
   (2) 报文鉴别：确认报文完整，包括确认发送者无误
2. 与加密是不同的概念
3. 与authorization也是不同的概念，授权目的是确认该过程是否被允许

1- 报文鉴别

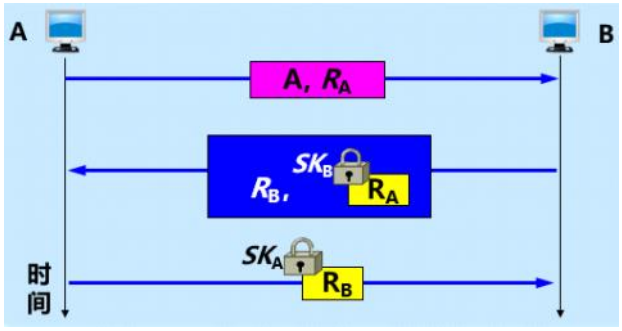
   (1) 数字签名给计算机很大负担，不常用于鉴别

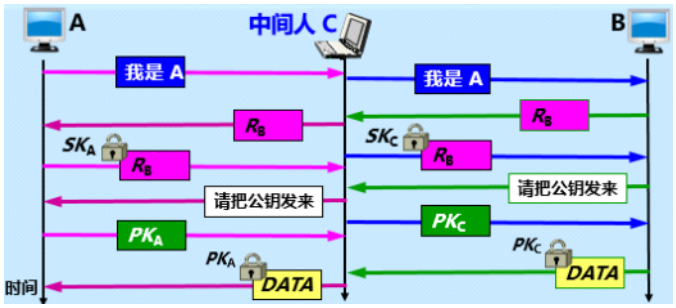
1. cryptographic hash function密码散列函数
   (1) 散列函数：对任意长的输入内容，输出一个长度固定的值
   (2) 散列函数是多对一的one-way单向函数，从散列函数值是可以伪造出伪原文的
   (3) 将固定位数的散列函数值拼接在报文后，方便检验，之前的checksum检验和就是这个思想的应用
2. Message Digest5报文摘要，RFC1321，1991年
   (1) 把二进制报文长度模2^64，求得64位余数，添加在报文后
   (2) 在报文后和长度余数间添加1个1和不超过511个0，使新长度为512倍数
   (3) 每512位，给报文分组一次，再给每组分为4个128位，再分成4个32位
   (4) 再给每小组算不同的散列函数值
   (5) 经王小云证明，一小时内能伪造出原文
3. Secure Hash Algorithm安全散列算法，稍慢但更安全
   (1) 按512位分组，算报文摘要值
   (2) 将分组和报文摘要值结合，产生报文摘要的下一个中间结果，直至完毕
   (3) 扫描5遍，使抗穷举性更高
4. Message Authentication Code报文鉴别码MAC
   (1) 只拼接一个散列函数值并不能仿伪造原文，最好对散列值再做一次加密

2- 实体鉴别

   (1) 报文鉴别对每个报文都要鉴别发送者
   (2) 实体鉴别只需在通信前验证一次

1. 最简单的实体鉴别：用只有双方知道的密钥做对称加密
   (1) replay attack重放攻击：截获A的报文，伪装成是C发出该报文的假象
   (2) IP欺骗：C假装自己的IP地址和A一样，让B彻底看不出C不是A

2. nonce不重数：不重复使用的大随机数，即一次一数
   (1) 通过每次重新对不重数用私钥加密，再塞入报文，应对重放攻击（中间人没有私钥，不能加密，相当于不能签名，实现鉴别）
   (2) 

3. man-in-the-middle attack中间人攻击
   (1) 

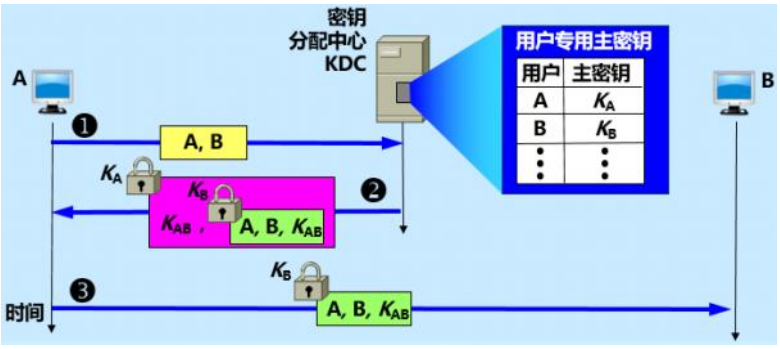
   (2) A 向 B 发送“我是 A”的报文，并给出了自己的身份。此报文被 “中间人” C 截获，C 把此报文原封不动地转发给 B。B 选择一个不重数 RB 发送给 A，但同样被 C 截获后也照样转发给 A
   (3) 中间人 C 用自己的私钥 SKC 对 RB 加密后发回给 B，使 B 误以为是 A 发来的。A 收到 RB 后也用自己的私钥 SKA 对 RB 加密后发回给 B，中途被 C 截获并丢弃。B 向 A 索取其公钥，此报文被 C 截获后转发给 A
   (4) C 把自己的公钥 PKC 冒充是 A 的发送给 B，而 C 也截获到 A 发送给 B 的公钥 PKA
   (5) B 用收到的公钥 PKC （以为是 A 的）对数据加密发送给 A。C 截获后用自己的私钥 SKC 解密，复制一份留下，再用 A 的公钥 PKA 对数据加密后发送给 A
   (6) A 收到数据后，用自己的私钥 SKA 解密，以为和 B 进行了保密通信。其实，B 发送给 A 的加密数据已被中间人 C 截获并解密了一份。但 A 和 B 却都不知道
      -
      - 密钥分配

1. 密钥管理：密钥的产生、分配、注入、验证、使用
   (1) 密钥分配/分发：网外分配、网内分配

1- 对称密钥的分配

1. 对称密钥网内分配难题
   (1) n个人互相通信，需要O（n方）的密钥数量
   (2) 互联网内分配密钥肯定需要加密，这个加密的密钥又如何分配
2. Key Distribution Center密钥分配中心KDC
   (1) 是一种大家都信任的机构，负责给想秘密通信的用户临时分配一个密钥

3. 用户A和B都是KDC事先登记的用户，已在KDC服务器事先分配到了与KDC通信时的master key主密钥KA和KB

4. 

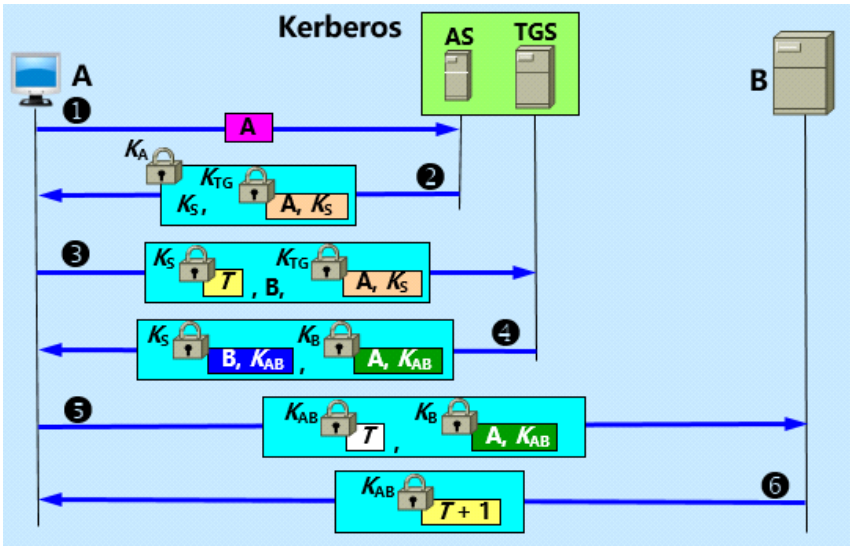
5. 为防止被重放攻击，还可在报文内加入时间戳

6. 主密钥KA和KB应定期更换，减少被破译的机会

1. Kerberos V5协议

    1) Kerbero同时是个KDC

    2) 使用AES加密

    3) 

    (1) A 用明文（包括登记的身份）向鉴别服务器 AS 表明自己的身份

    (2) AS 向 A 发送用 A 的对称密钥 KA 加密的报文，这个报文包含 A 和 TGS 通信的会话密钥 KS，以及 AS 要发送给 TGS 的票据（这个票据是用 TGS 的对称密钥 KTG 加密的）

    (3) A 向 TGS 发送三个项目：

        1) 转发鉴别服务器 AS 发来的票据

        2) 服务器 B 的名字。这表明 A 请求 B 的服务。请注意，现在 A 向 TGS 证明自己的身份并非通过键入口令（因为入侵者能够从网上截获明文口令），而是通过转发 AS 发出的票据（只有 A 才能提取出）。票据是加密的，入侵者伪造不了

        3) 用 KS 加密的时间戳 T 。它用来防止入侵者的重放攻击

    (4) TGS 发送两个票据，每一个都包含 A 和 B 通信的会话密钥 KAB 。给 A 的票据用 KS 加密；给 B 的票据用 B 的密钥 KB 加密。请注意，现在入侵者不能提取 KAB，因为不知道 KA 和 KB 。入侵者也不能重放步骤（3），因为入侵者不能把时间戳更换为一个新的（因为不知道 KS ）

    (5) A 向 B 转发 TGS 发来的票据，同时发送用 KAB 加密的时间戳 T

1) 为防止被重放，Kerberos要求所有主机在时钟上松散的同步，即误差不要超过5分钟

(6) B 用时间戳 T+1 来证实收到了票据。B 向 A 发送的报文用密钥KAB 加密

1) 以后，A 和 B 就使用 TGS 给出的会话密钥 KAB 进行通信

1- 公钥的分配

1. Certification Authority认证中心CA：将公钥与其对应实体进行binding绑定
   (1) 一般是政府出资建立的机构
   (2) 任何用户都能从可信的地方获得CA的公钥，验证某个公钥是否被某个实体拥有
2. 每个实体都有CA发来的certificate证书，里面有公钥及用户标识
   (1) 此证书被CA数字签名了，不可伪造
   (2) ITU-T制定了X.509协议标准，描述整数结构
   (3) IETF对X.509做出了少量改动，给出了Public Key Infrastructure公钥及出结构PKI

版本号	区分 X.509 不同版本
序列号	CA 发放，唯一
签名算法	签署证书所使用的算法和参数
发行者	CA 的 X.509 名字
有效期	包括起始时间和终止时间
主体名	证书持有者的名称及有关信息
公钥	有效的公钥及其使用方法
发行者 ID	任选，唯一，标识发行者
主体 ID	任选，唯一，标识证书持有者
扩展域	扩充信息
认证机构签名	用 CA 私钥对证书签名

3. CA证书过期或吊销
   (1) 用户私钥被泄漏
   (2) 用户不再被CA认证
   (3) CA前述用户证书的私钥被泄漏
   (4) 以上三种情况会使CA证书被吊销，CA需建立并维护一个证书吊销列表
      1)
      2)
      3)
      4)
      5) ----------我是底线----------